Advanced Predictive Modeling

Dr. Bowen Hui
Computer Science
University of British Columbia Okanagan

Plan for Upcoming Lectures

- Lecture 1:
  - Regression -> classification -> Bayesian networks
- Lecture 2 (this class):
  - Rational decision making
- Lecture 3:
  - Sequential decision making

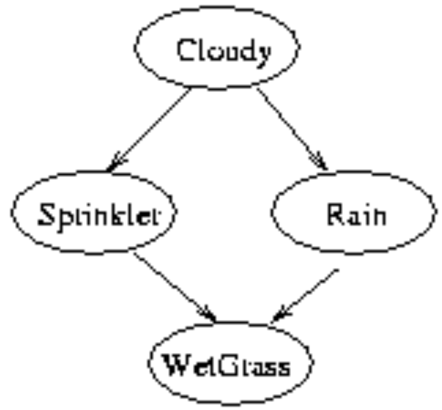
Recall: Bayesian Network (BN)

- Graphical models provide visual explanatory power in modeling real world problems
- BN is a directed acyclic graph that represents causality among a set of random variables
  - Nodes: random variables, each with an associated probability function
  - Edges: represents conditional dependencies between nodes
- Specifies a complete joint probability distribution over all the variables in the model
- Answers possible inference queries by marginalization

Recall: Weather Example

Pr(C):

P(C=F)	P(C=T)
0.5	0.5



Pr(S|C):

C	P(S=F)	P(S=T)
F	0.5	0.5
T	0.9	0.1

Pr(R|C):

C	P(R=F)	P(R=T)
F	0.8	0.2
T	0.2	0.8

Pr(W|S,R):

S	R	P(W=F)	P(W=T)
F	F	1.0	0.0
T	F	0.1	0.9
F	T	0.1	0.9
T	T	0.01	0.99

Complete joint distr.:

Pr(C,S,R,W) = Pr(C) * Pr(S|C) * Pr(R|C) * Pr(W|S,R)

How to Make Decisions?

- Choose an action
  - Estimate future consequences
  - Based on our beliefs of the world

- Bayes nets enable us to maintain beliefs of the world via probabilistic inference

- *Next*: modeling consequences and selecting actions

Probabilistic Inference

- Formally:
  - Given a prior distribution *Pr* over some variables (represents degrees of belief over variables)
  - Given new evidence $E = e$ for some variable E
  - Revise your degrees of belief to get the posterior distribution, Pr_e

Probabilistic Inference

- Formally:
  - Given a prior distribution *Pr* over some variables (represents degrees of belief over variables)
  - Given new evidence $E = e$ for some variable E
  - Revise your degrees of belief to get the posterior distribution, Pr_e

- Intuition:
  - How do your degrees of belief change as a result of learning $E = e$?
    (or more generally, $\mathbf{E} = \mathbf{e}$, for set $\mathbf{E}$)

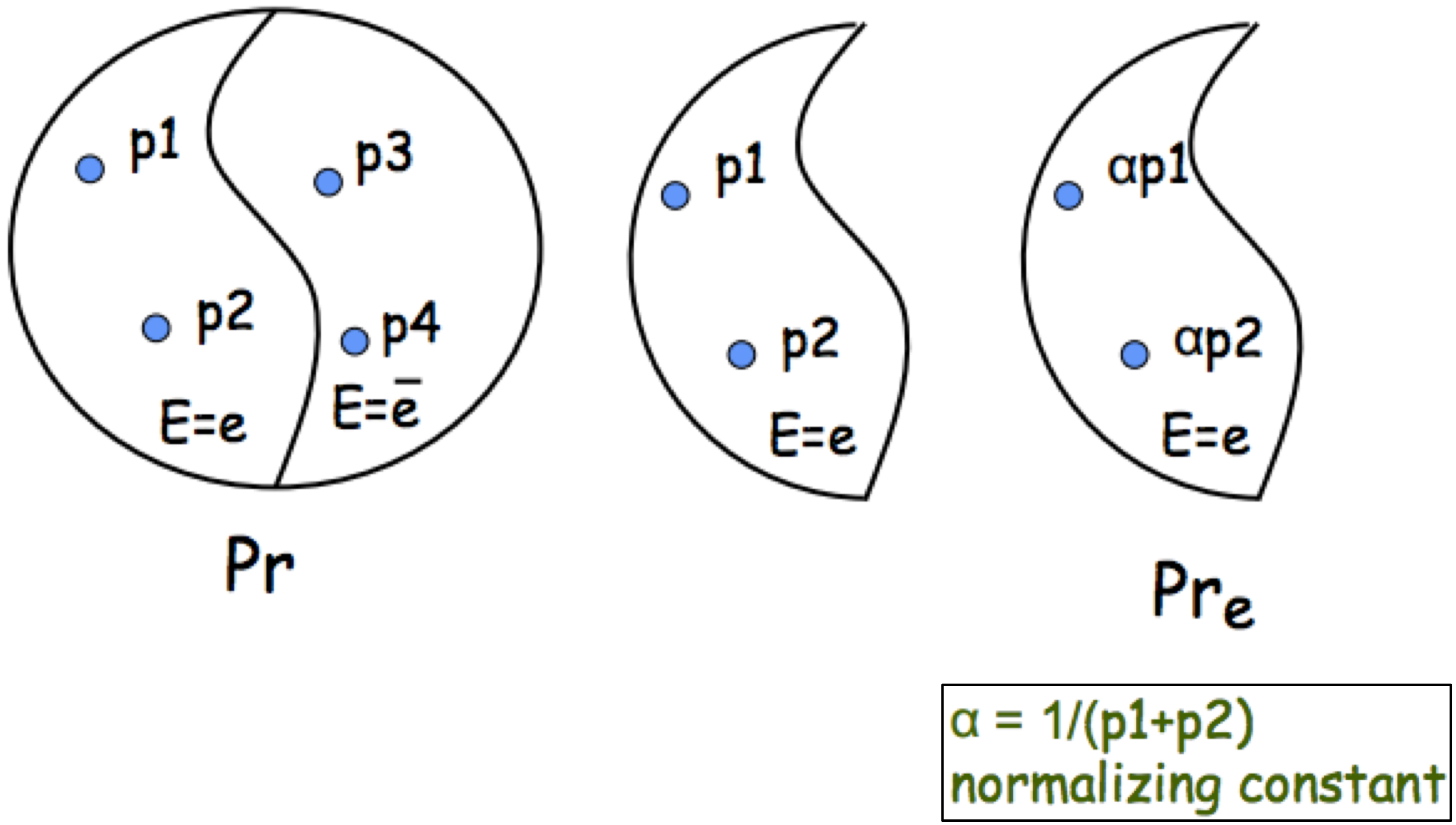
Conditioning

- We define:

$$Pr_e(a) = Pr(a|e)$$

- That is, we produce Pr_e by conditioning the prior distribution on the observed evidence e

Semantics of Conditioning



Computational Bottleneck

- How do we specify the full joint distribution over a set of RVs $X_1, \ldots, X_n$?

- Inference in this representation is frightfully slow

Computational Bottleneck

- How do we specify the full joint distribution over a set of RVs $X_1, \ldots, X_n$?
  - Exponential number of possible worlds
  - These numbers are not robust/stable
  - These numbers are not natural to assess

- Inference in this representation is frightfully slow

Computational Bottleneck

- How do we specify the full joint distribution over a set of RVs $X_1, \ldots, X_n$?

- Inference in this representation is frightfully slow
  - Must sum over exponential number of worlds to answer query *Pr(a)* or to condition on evidence *e* to determine $Pr_e(a)$

Consider Headache Example

sunny

	cold	~cold
headache	0.108	0.012
~headache	0.016	0.064

~sunny

	cold	~cold
headache	0.072	0.008
~headache	0.144	0.576

$$\Pr(headache) = 0.108 + 0.012 + 0.072 + 0.008 = 0.2$$

$$\Pr(headache \wedge cold | sunny) = \Pr(headache \wedge cold \wedge sunny) \,/\, \Pr(sunny)$$
$$= \; 0.108/(0.108 + 0.012 + 0.016 + 0.064) = 0.54$$

$$\Pr(headache \wedge cold | {\sim}sunny) = \Pr(headache \wedge cold \wedge {\sim}sunny) \,/\, \Pr({\sim}sunny)$$
$$= \; 0.072/(0.072 + 0.008 + 0.144 + 0.576) = 0.09$$

Practical Solution

- How to avoid these two bottlenecks?
  - No solution in general
  - In practice, we will exploit structure

- Use independence and conditional independence assumptions

The Value of Independence

- Complete independence reduces both representation of joint distribution and inference from $O(2^n)$ to $O(n)$
- Unfortunately, complete independence is very rare
  - Most realistic domains don’t exhibit this property
- Fortunately, most domains exhibit a fair amount of conditional independence
  - Can exploit conditional independence for representation and inference too
  - Bayesian networks do just this

Exploiting Conditional Independence

- Consider the following story:
  - If Bowen woke up too early (E), she needs caffeine (C)
  - If Bowen needs caffeine, she's likely to be grumpy (G)
  - If she is grumpy, then her lecture won't be as good (L)
  - If the lecture doesn't go smoothly, then students will be disappointed (S)

E = Woke up too early
C = need caffeine
G = gets grumpy
L = lecture not smooth
S = students disappointed

Conditional Independence

- If you learned any of E,C,G,L, would your assessment of Pr(S) change?

E = Woke up too early

C = need caffeine

G = gets grumpy

L = lecture not smooth

S = students disappointed

Conditional Independence

- If you learned any of E,C,G,L, would your assessment of Pr(S) change?
  – If any of E,C,G,L are true, you would increase Pr(s) and decrease Pr(~s)
  – Therefore, S is <u>not</u> independent of E,C,G,L

E = Woke up too early
C = need caffeine
G = gets grumpy
L = lecture not smooth
S = students disappointed

Conditional Independence

- If you knew the value of L (true or false), would learning the value of E,C, or G influence your assessment of Pr(S)?

E = Woke up too early

C = need caffeine

G = gets grumpy

L = lecture not smooth

S = students disappointed

Conditional Independence

- If you knew the value of L (true or false), would learning the value of E,C, or G influence your assessment of Pr(S)?
  - Influence that E,C,G has on S is mediated by L
  - E.g. Students aren’t disappointed because Bowen is grumpy, it’s because the lecture wasn’t smooth
  - So S is independent of E,C,G, <u>given</u> L

E = Woke up too early
C = need caffeine
G = gets grumpy
L = lecture not smooth
S = students disappointed

Conditional Independence

- We have: S is independent of E,C,G, given L
- Similarly:
  - L is independent of E,C given G
  - G is independent of E given C
- This translates to:
  - Pr(S|L,G,C,E) = Pr(S|L)
  - Pr(L|G,C,E) = Pr(L|G)
  - Pr(G|C,E) = Pr(G|C)
  - Pr(C|E) % doesn't simplify further
  - Pr(E) % doesn't simplify further

E = Woke up too early
C = need caffeine
G = gets grumpy
L = lecture not smooth
S = students disappointed

Conditional Independence

- Specifying the full joint distribution Pr(S,L,G,C,E):

E = Woke up too early

C = need caffeine

G = gets grumpy

L = lecture not smooth

S = students disappointed

Conditional Independence

- Specifying the full joint distribution Pr(S,L,G,C,E):
- By the chain rule:
  Pr(S,L,G,C,E) =
  Pr(S|L,G,C,E)Pr(L|G,C,E)Pr(G|C,E)Pr(C|E)Pr(E)

E = Woke up too early
C = need caffeine
G = gets grumpy
L = lecture not smooth
S = students disappointed

Conditional Independence

- Specifying the full joint distribution Pr(S,L,G,C,E):

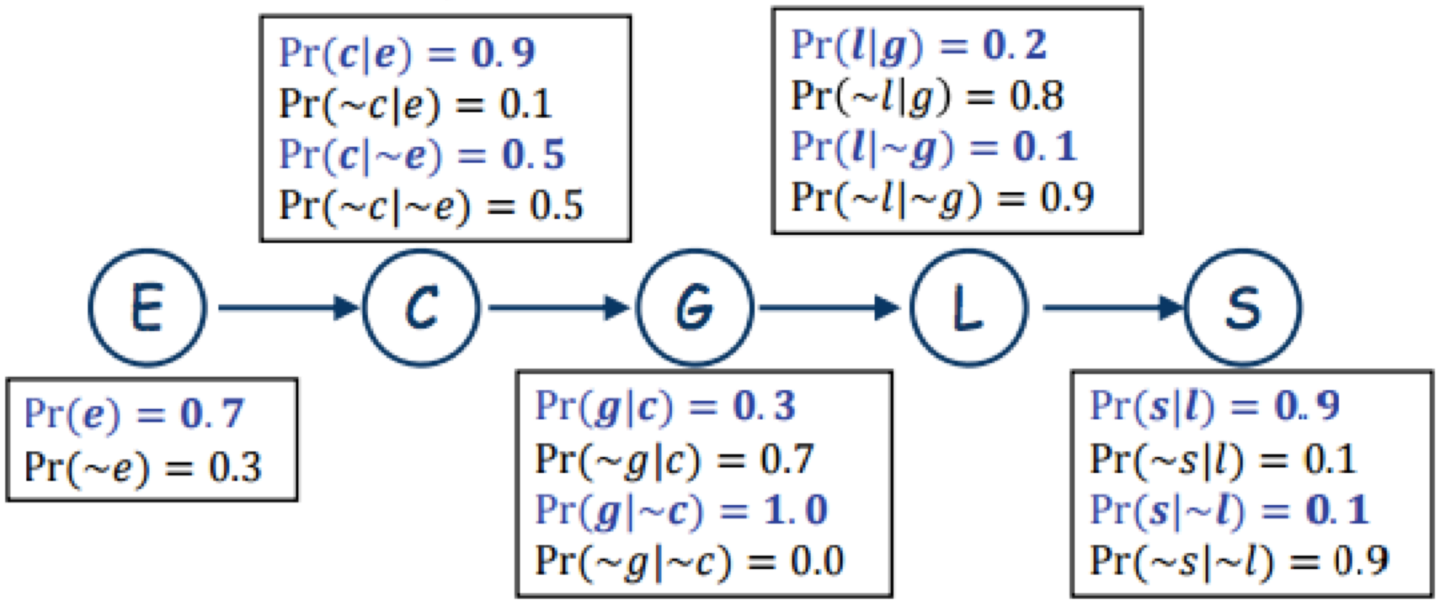
- By the chain rule:
  Pr(S,L,G,C,E) =
  Pr(S|L,G,C,E)Pr(L|G,C,E)Pr(G|C,E)Pr(C|E)Pr(E)

- By our independence assumptions:
  Pr(S,L,G,C,E) = Pr(S|L)Pr(L|G)Pr(G|C)Pr(C|E)Pr(E)

- The full joint is specified by 5 local conditional distributions!

E = Woke up too early
C = need caffeine
G = gets grumpy
L = lecture not smooth
S = students disappointed

Example Quantification



- Specifying the joint requires only 9 parameters!
  - Instead of 31 (= 2^5 – 1) for explicit representation

Inference is Easy

- How to compute Pr(g)?

E = Woke up too early

C = need caffeine

G = gets grumpy

L = lecture not smooth

S = students disappointed

Inference is Easy

- How to compute Pr(g)?
  - Apply the sum-out rule
- $$P(g) = \sum_{c_i \in Dom(C)} \Pr(g \mid c_i) \Pr(c_i)$$

$$= \sum_{c_i \in Dom(C)} \Pr(g \mid c_i) \sum_{e_i \in Dom(E)} \Pr(c_i \mid e_i) \Pr(e_i)$$

Terms available in our local distributions!

E = Woke up too early
C = need caffeine
G = gets grumpy
L = lecture not smooth
S = students disappointed

Inference is Easy

- Concrete example to compute Pr(g):

$$Pr(c) = Pr(c|e)Pr(e) + Pr(c|{\sim}e)Pr({\sim}e)$$
$$= 0.8 * 0.7 + 0.5 * 0.3 = 0.78$$

$$Pr({\sim}c) = 1 - Pr(c)$$
$$= 0.22$$

$$Pr(g) = Pr(g|c)Pr(c) + Pr(g|{\sim}c)Pr({\sim}c)$$
$$= 0.3 * 0.78 + 1.0 * 0.22 = 0.454$$

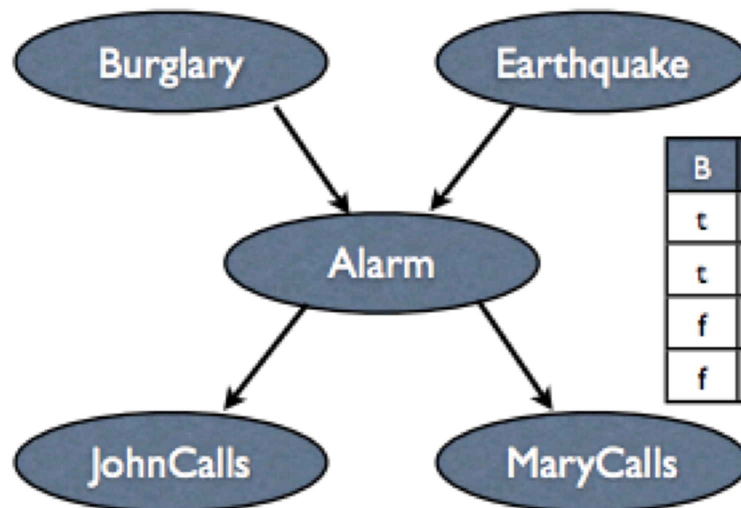
E = Woke up too early
C = need caffeine
G = gets grumpy
L = lecture not smooth
S = students disappointed

BN Example #3

- Burglary network (from J. Pearl)
  - Burglary = burglary occurs at your house
  - Earthquake = earthquake occurs at your house
  - Alarm = alarm goes off
  - JohnCalls = John calls to report the alarm
  - MaryCalls = Mary calls to report the alarm

P(B)
0.001

P(E)
0.002



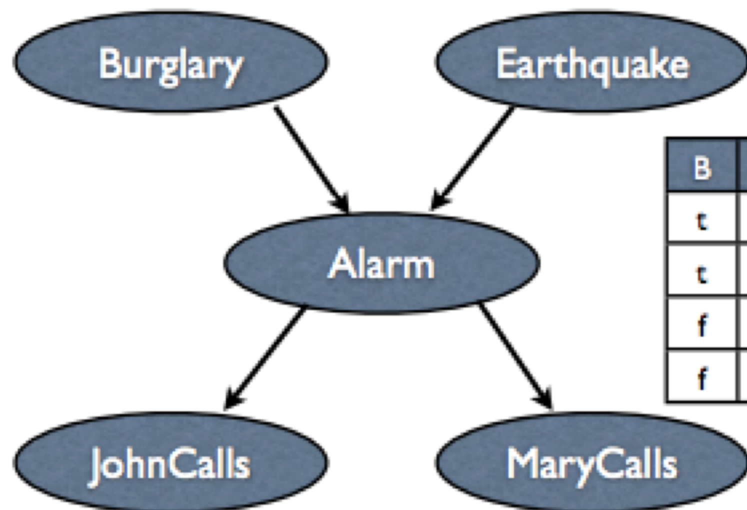
B	E	P(A|B,E)
t	t	0.95
t	f	0.94
f	t	0.29
f	f	0.001

A	P(JC|A)
t	0.90
f	0.05

A	P(MC|A)
t	0.70
f	0.01

BN Example #3

- Compute an entry in the joint distribution:
  Pr(B=t, E=f, A=t, MC=t, JC=f)
  = ?



P(B)
0.001

P(E)
0.002

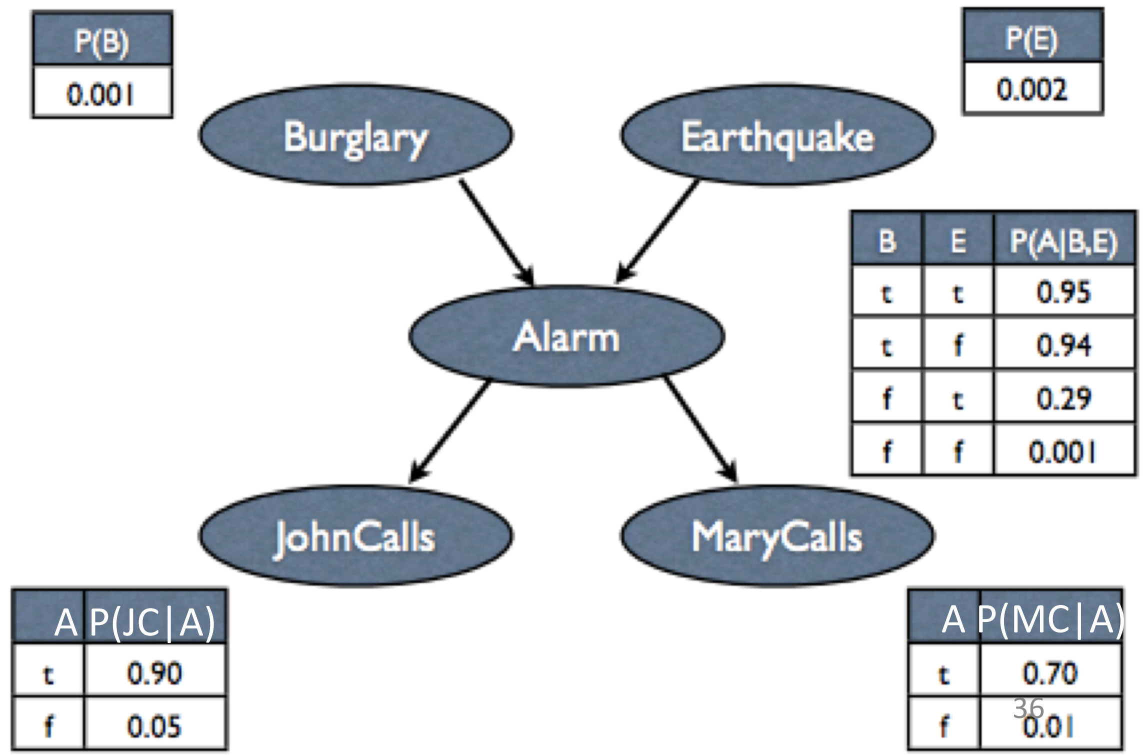
B	E	P(A|B,E)
t	t	0.95
t	f	0.94
f	t	0.29
f	f	0.001

A	P(JC|A)
t	0.90
f	0.05

A	P(MC|A)
t	0.70
f	0.01

BN Example #3

- Compute an entry in the joint distribution:
  Pr(B=t, E=f, A=t, MC=t, JC=f)
  = Pr(B=t)Pr(E=f)Pr(A=t|B=t,E=f)Pr(MC=t|A=t)Pr(JC=f|A=t)
  = 0.001 * (1-0.002) * (0.94) * (0.70) * (1-0.90)
  = 0.0000656684

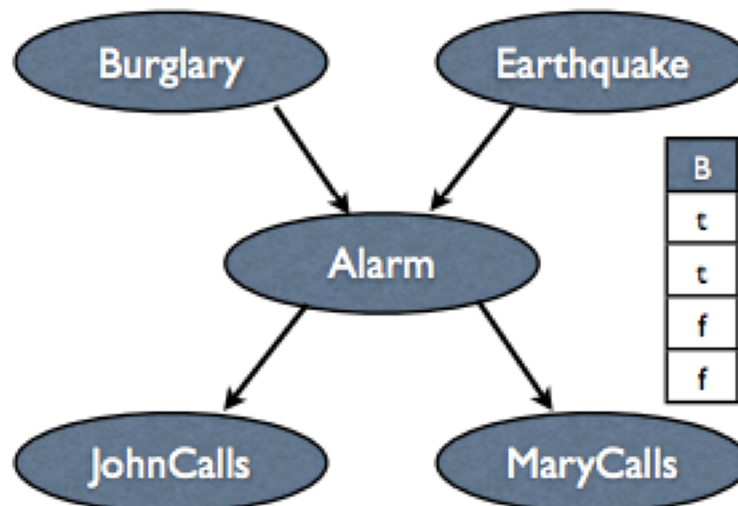


BN Example #3

- Compute the marginal probability that Mary calls
  Pr(MC=t) = ?

P(B)
0.001

P(E)
0.002



B	E	P(A|B,E)
t	t	0.95
t	f	0.94
f	t	0.29
f	f	0.001

A	P(JC|A)
t	0.90
f	0.05

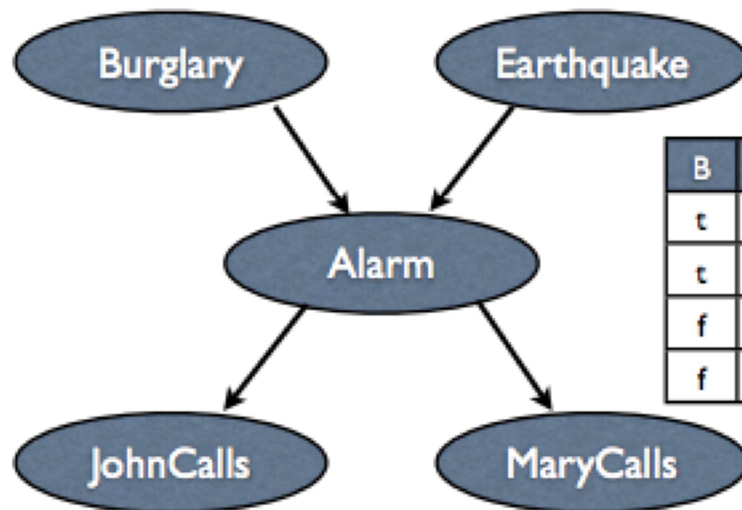
A	P(MC|A)
t	0.70
f	0.01

BN Example #3

- Compute the marginal probability that Mary calls
  Pr(MC=t) = $\sum_{B,E,A,JC} \Pr(MC = t, B, E, A, JC)$ = …

P(B)
0.001

P(E)
0.002



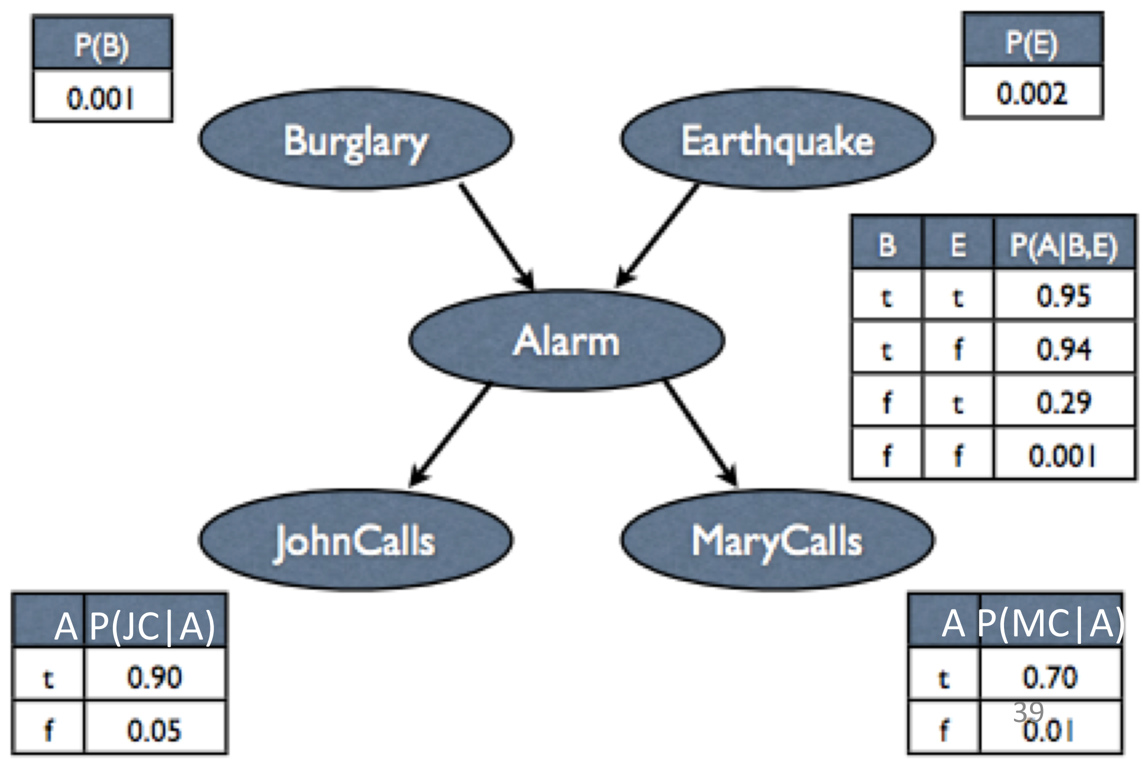
B	E	P(A|B,E)
t	t	0.95
t	f	0.94
f	t	0.29
f	f	0.001

A	P(JC|A)
t	0.90
f	0.05

A	P(MC|A)
t	0.70
f	0.01

BN Example #3

- Making a prediction given evidence upstream in the graph
- E.g. Pr(MC=t | B=t) = ?



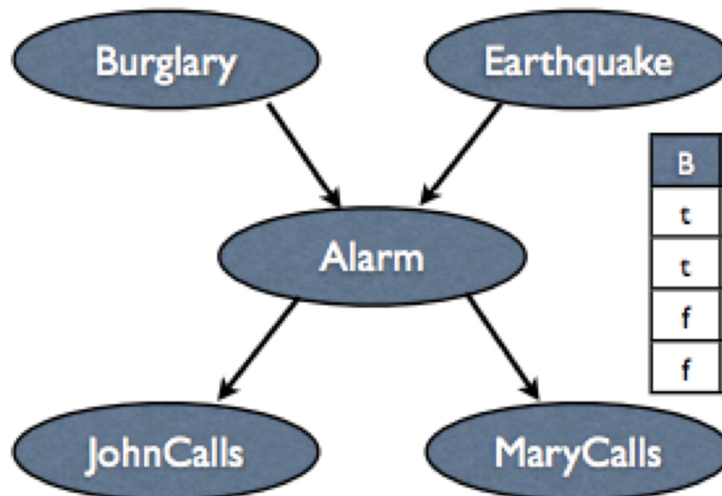
BN Example #3

- Making a prediction given evidence upstream in the graph
- E.g. Pr(MC=t | B=t)

$$= \frac{\Pr(MC=t, B=t)}{\Pr(B=t)} = \frac{\sum_{E,A,JC} \Pr(MC = t, B = 1, E, A, JC)}{\sum_{E,A,JC,MC} \Pr(B = t, E, A, JC, MC)} = \ldots$$

P(B)
0.001

P(E)
0.002



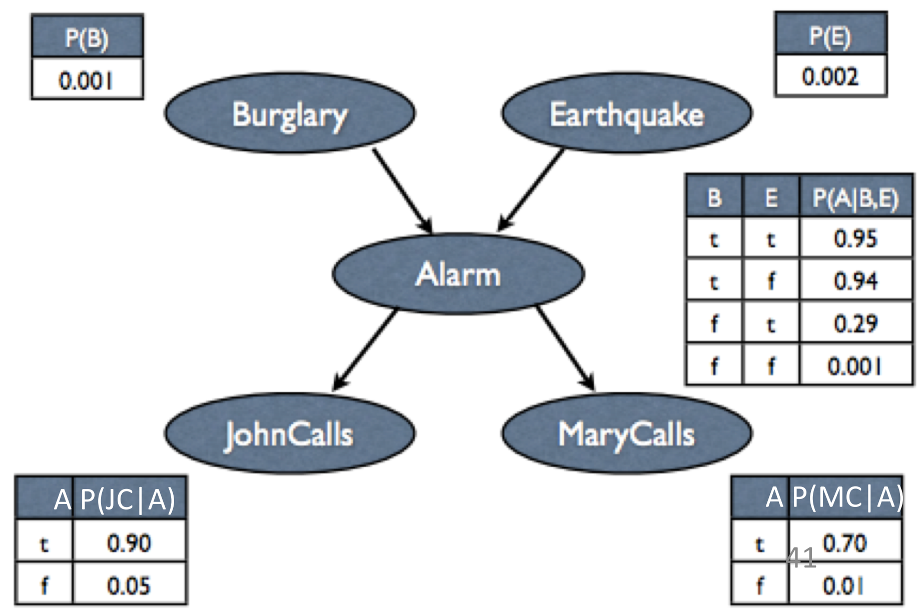
B	E	P(A|B,E)
t	t	0.95
t	f	0.94
f	t	0.29
f	f	0.001

A	P(JC|A)
t	0.90
f	0.05

A	P(MC|A)
t	0.70
f	0.01

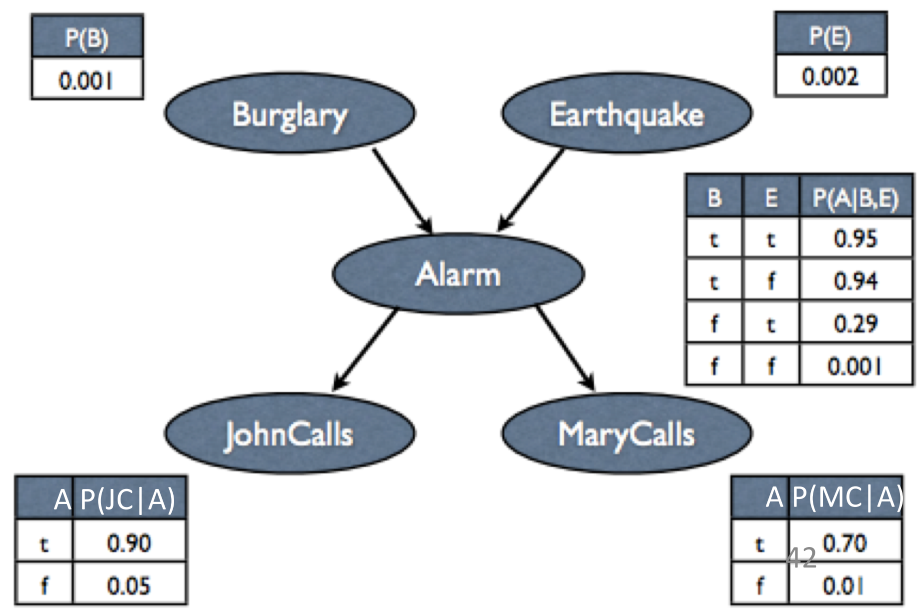
BN Example #3

- Finding an explanation given evidence downstream in the graph
- Pr(B=t | MC=t) = ?
- Pr(E=t | MC=t) = ?



BN Example #3

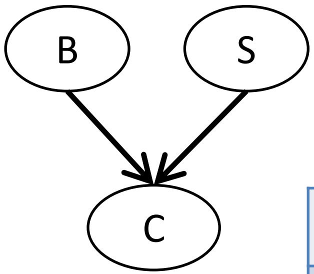
- Finding an explanation given evidence downstream in the graph
- $$\Pr(B{=}t \mid MC{=}t) = \frac{\Pr(MC{=}t \mid B{=}t)\Pr(B{=}t)}{\Pr(MC{=}t)} = \ldots$$
- $$\Pr(E{=}t \mid MC{=}t) = \frac{\Pr(MC{=}t \mid E{=}t)\Pr(E{=}t)}{\Pr(MC{=}t)} = \ldots$$



BN Example #4

- Suppose a college admits students who are either brainy (B) or sporty (S) or both.
- Let C denote the event that someone is admitted to college.
- Suppose in the general population that B and S are independent.

Pr(~B)	Pr(B)
0.5	0.5



Pr(~S)	Pr(S)
0.5	0.5

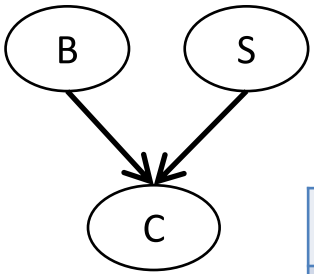
		Pr(~C|B,S)	Pr(C|B,S)
~B	~S	1	0
B	~S	0	1
~B	S	0	1
B	S	0	1

BN Example #4

- As it turns out, looking at population of college students (those for which C is observed to be true)
- It will be found that being brainy makes you less likely to be sporty and vice versa!
- Because either property alone is sufficient to explain the evidence on C

Pr(~S)	Pr(S)
0.5	0.5

Pr(~B)	Pr(B)
0.5	0.5

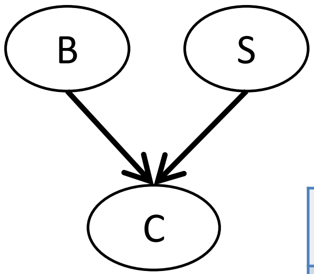


		Pr(~C|B,S)	Pr(C|B,S)
~B	~S	1	0
B	~S	0	1
~B	S	0	1
B	S	0	1

BN Example #4

- i.e. Pr(S=true | C=true, B=true) $\leq$ Pr(S=true | C=true) and Pr(B=true | C=true, S=true) $\leq$ Pr(B=true | C=true)
- This is known as the explaining away phenomenon
  - Happens when two causes "compete" to explain the observed data

Pr(~B)	Pr(B)
0.5	0.5



Pr(~S)	Pr(S)
0.5	0.5

		Pr(~C|B,S)	Pr(C|B,S)
~B	~S	1	0
B	~S	0	1
~B	S	0	1
B	S	0	1

Implementing BN Example #4

```
B = 1; S = 2; C = 3;
dag = zeros(3,3);
dag([B S], C)=1;
ns = 2*ones(1,3);
bnet = mk_bnet(dag, ns);   % makes the BN structure

% populate CPTs
bnet.CPD{B} = tabular_CPD(bnet, B, 'CPT', [0.5 0.5]');
bnet.CPD{S} = tabular_CPD(bnet, S, 'CPT', [0.5 0.5]');
CPT = zeros(2,2,2);
CPT(1,1,:) = [1 0];
CPT(2,1,:) = [0 1];
CPT(1,2,:) = [0 1];
CPT(2,2,:) = [0 1];
bnet.CPD{C} = tabular_CPD(bnet, C, 'CPT', CPT);
```

Inference with BN Example #4

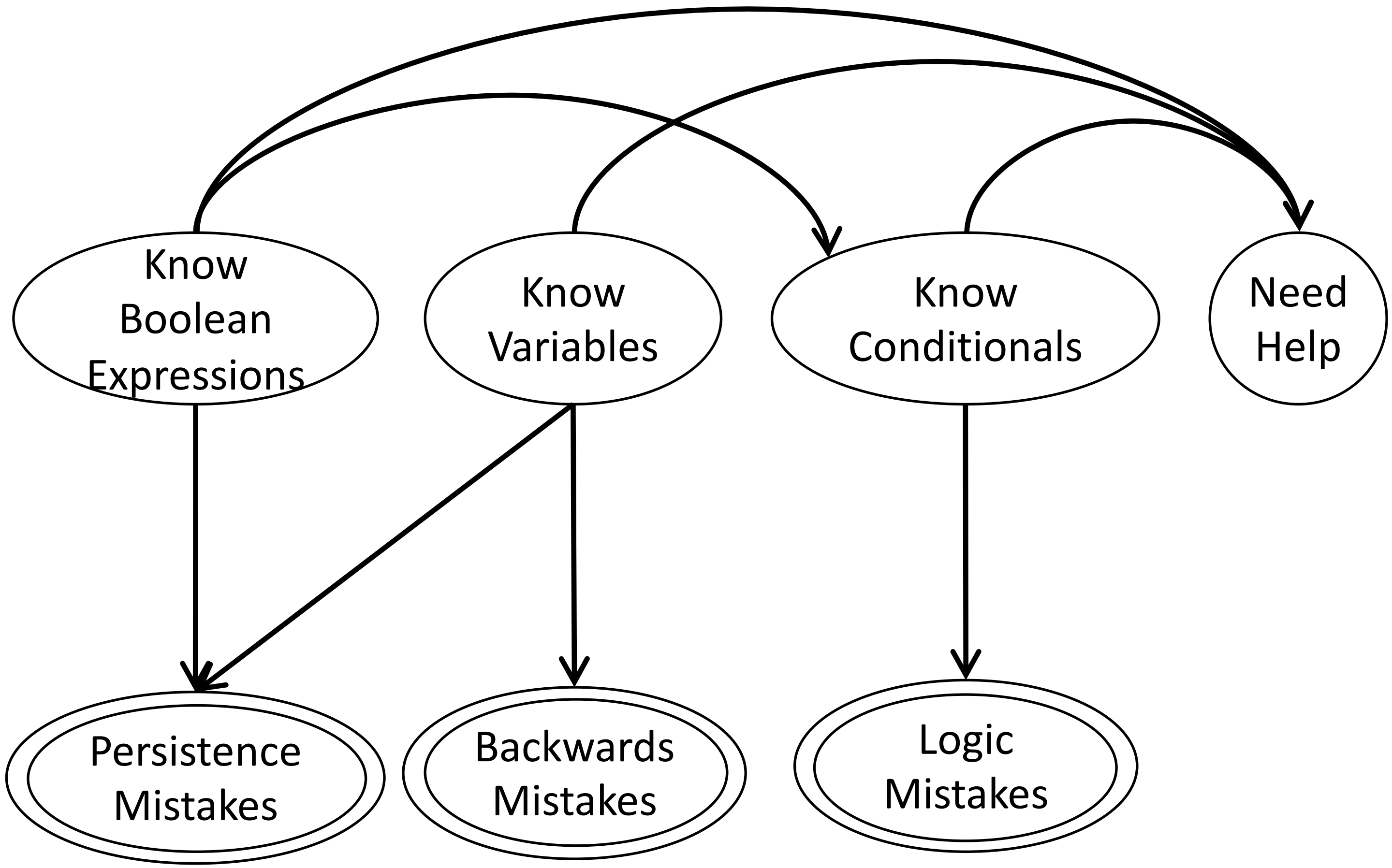
```
% matlab convention: 1=false 2=true
engine = jtree_inf_engine(bnet);
ev = cell(1,3);

ev{C} = 2;    % C is observed to be true
engine = enter_evidence(engine, ev);
m = marginal_nodes(engine, B);
fprintf('P(B=true|C=true) = %5.3f\n', m.T(2))

ev{S} = 2;    % C and S are now true
engine = enter_evidence(engine, ev);
m = marginal_nodes(engine, B);
fprintf('P(B=true|C=true,S=true) = %5.3f\n', m.T(2))
```

The General Inference Task

- Given:
  - Query variables: **X**
  - Evidence (observed) variables: **E = e**
  - Unobserved variables: **Y**

- Goal: To calculate useful information about the query variables
  - Updating belief of **X**: Pr(**X**|**e**)
  - Most probable explanation (MPE): $\mathbf{x}^* = \text{argmax}_{\mathbf{x}} \Pr(\mathbf{x}|\mathbf{e})$

Which Task Is It?



How likely is the student to need help?

a) Belief update: $\Pr(\mathbf{X}|\mathbf{e})$

b) Most probable explanation: $\mathbf{x}^* = \text{argmax}_{\mathbf{x}} \Pr(\mathbf{x}|\mathbf{e})$

The General Inference Task

- Given:
  - Query variables: **X**
  - Evidence (observed) variables: **E = e**
  - Unobserved variables: **Y**

- Goal: To calculate useful information about the query variables
  - Updating belief of **X**: Pr(**X**|**e**)
  - Most probable explanation (MPE): $\mathbf{x}^* = \text{argmax}_{\mathbf{x}} \Pr(\mathbf{x}|\mathbf{e})$

- Recall inference via the full joint distribution:

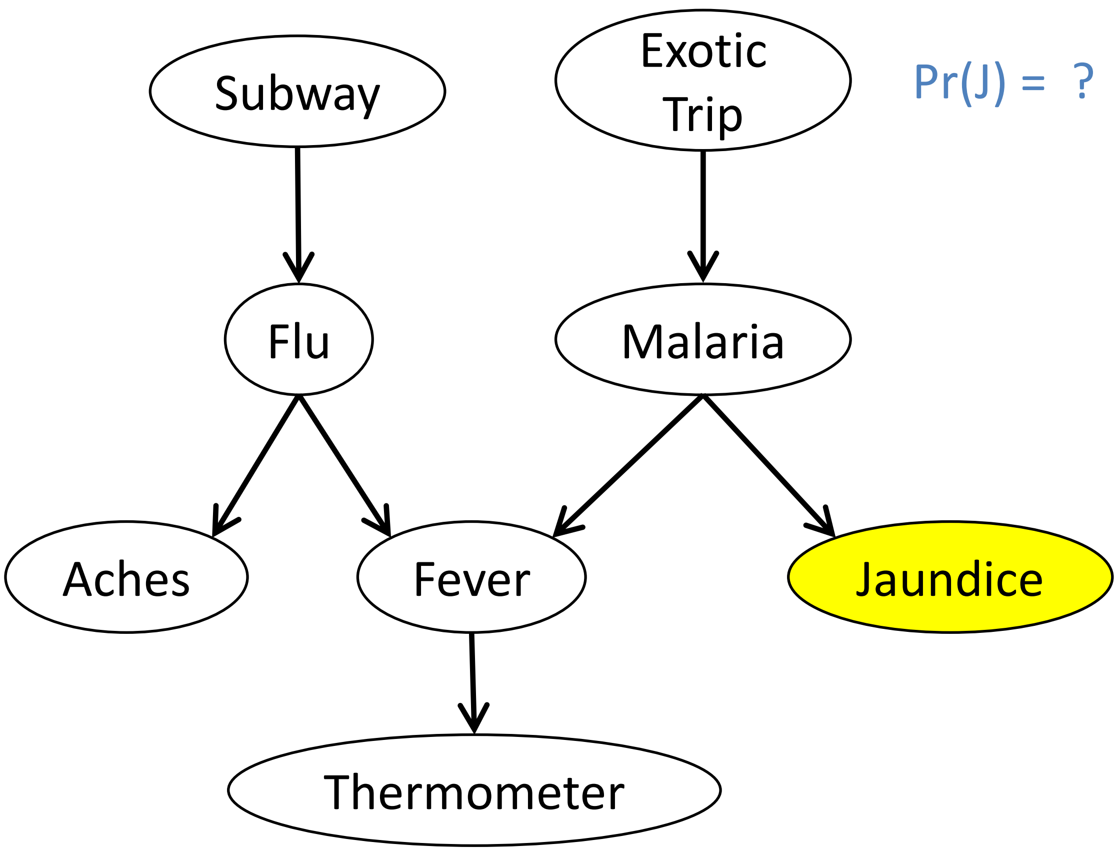
$$\Pr(\boldsymbol{X}|\boldsymbol{E}=\boldsymbol{e}) = \frac{\Pr(\boldsymbol{X},\boldsymbol{e})}{\Pr(\boldsymbol{e})} \propto \sum_{Y} \Pr(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{e})$$

Inference Algorithms

- Purpose:
  - Understand the intuition behind the computation steps involved in an inference algorithm
- Need to know:
  - Given a problem, know how to formulate the query for inference task
    - Which are your query variable(s)?
    - Which are your evidence variable(s)?
  - Given a problem, know if you are doing belief update or MPE

Simple Forward Inference

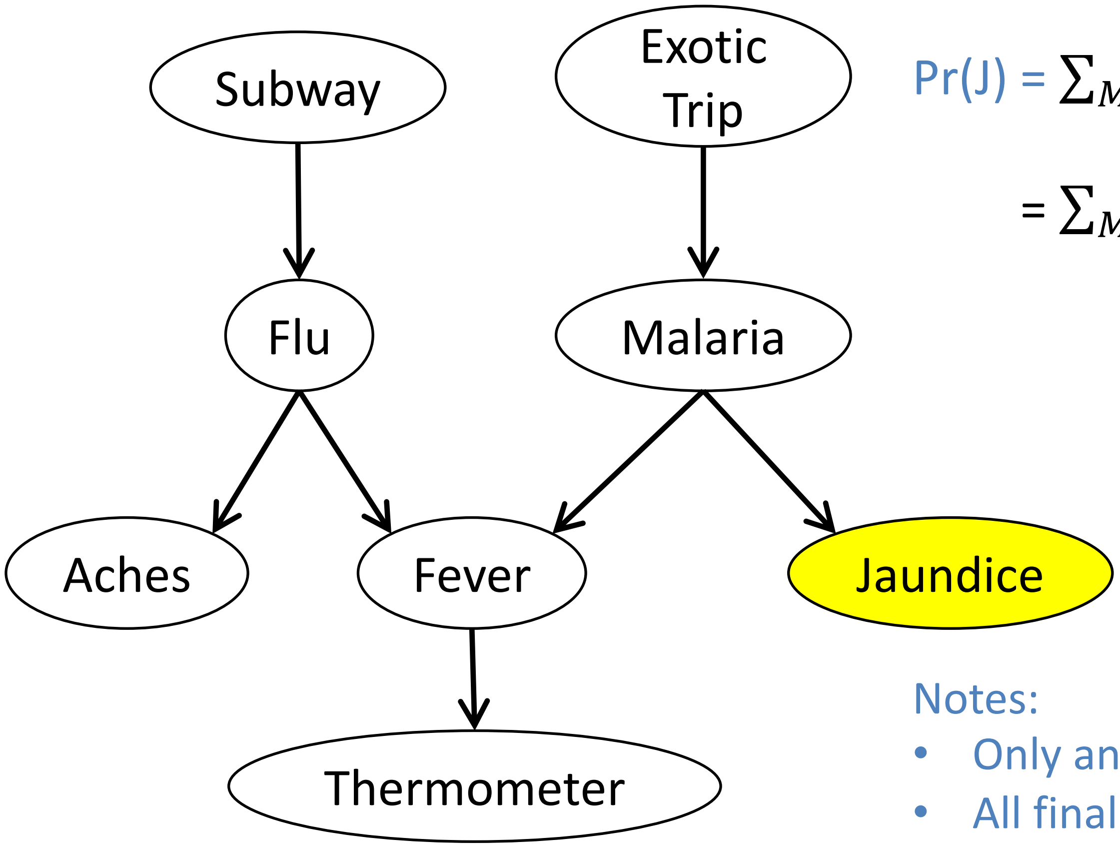
- Computing marginal requires simple forward "propagation" of probabilities



Pr(J) = ?

Simple Forward Inference

- Computing marginal requires simple forward "propagation" of probabilities



Pr(J) = $\sum_{M,ET} \Pr(J|M) \Pr(M|ET) \Pr(ET)$

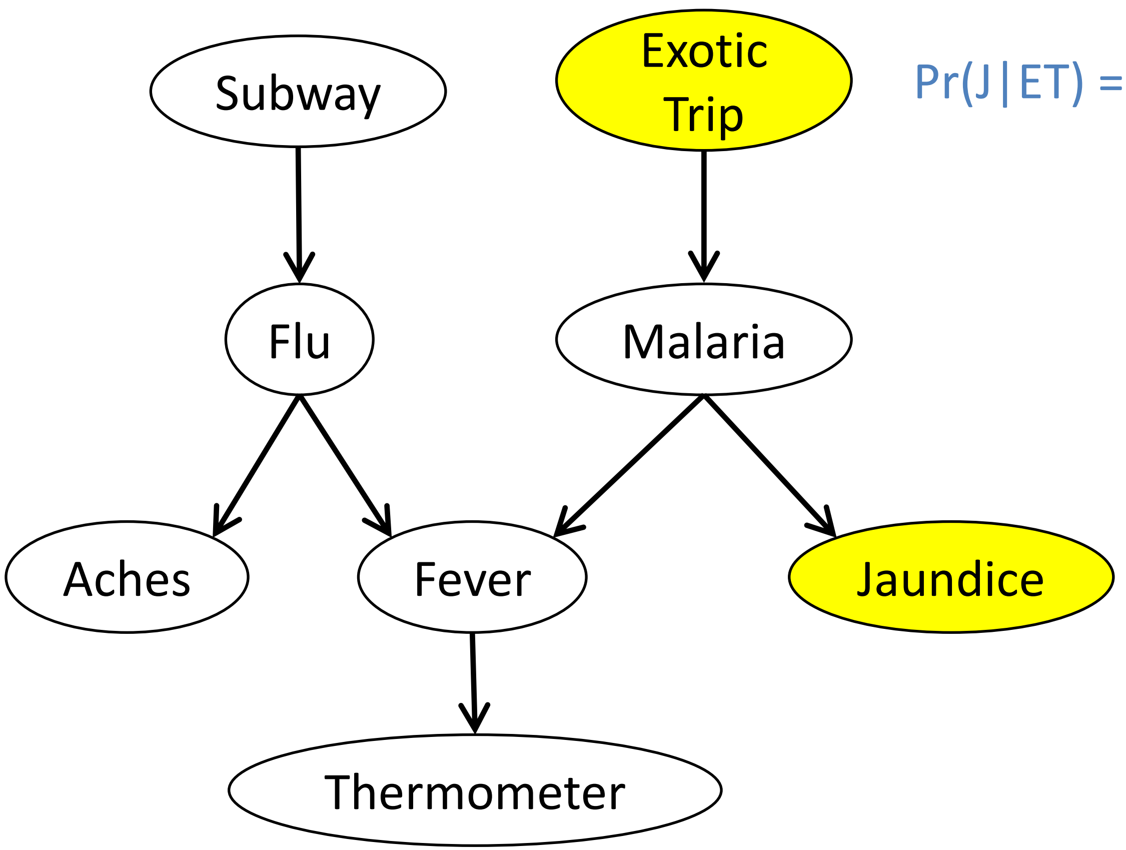
= $\sum_{M} \Pr(J|M) \sum_{ET} \Pr(M|ET) \Pr(ET)$

Notes:

- Only ancestors of J considered
- All final terms are CPTs in the BN

Simple Forward Inference

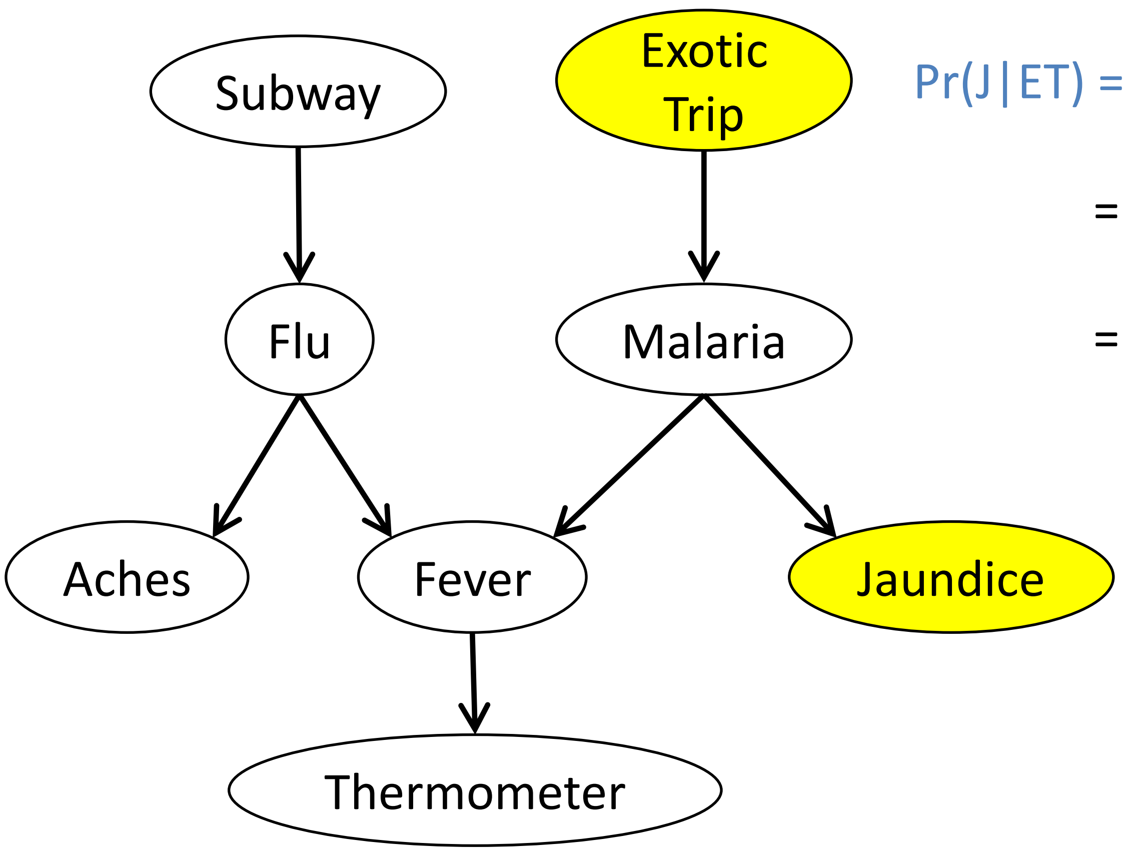
- Same idea applies when we have upstream evidence



Pr(J|ET) = ?

Simple Forward Inference

- Same idea applies when we have upstream evidence



$$\Pr(J|ET) = \sum_M \Pr(J, M|ET)$$

$$= \sum_M \Pr(J|M, ET)\Pr(M|ET)$$

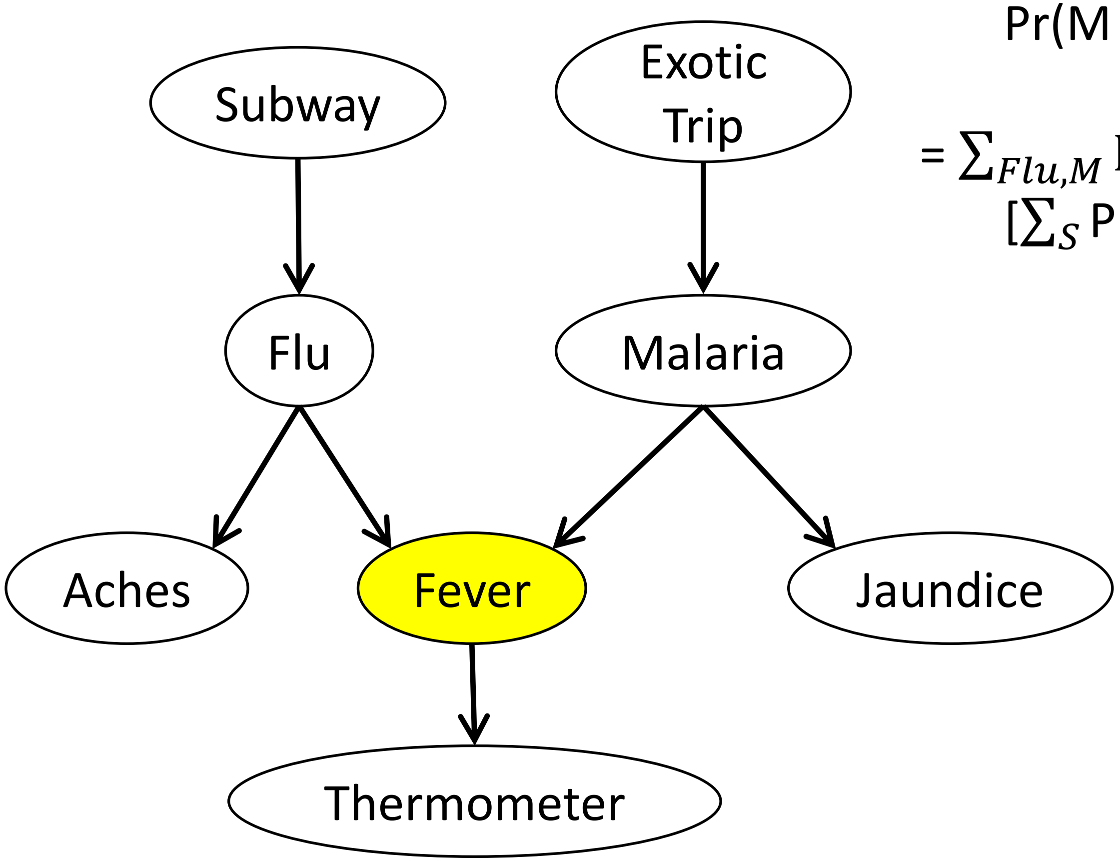
$$= \sum_M \Pr(J|M)\Pr(M|ET)$$

Simple Forward Inference

- Same idea applies with multiple parents

$$\Pr(\text{Fever}) = \sum_{Flu,M,S,ET} \Pr(Fever|Flu,M)\Pr(Flu|S)\Pr(M|ET)\Pr(S)\Pr(ET)$$

$$= \sum_{Flu,M} \Pr(Fever|Flu,M)\left[\sum_{S} \Pr(Flu|S)\Pr(S)\right]\left[\sum_{ET} \Pr(M|ET)\Pr(ET)\right]$$



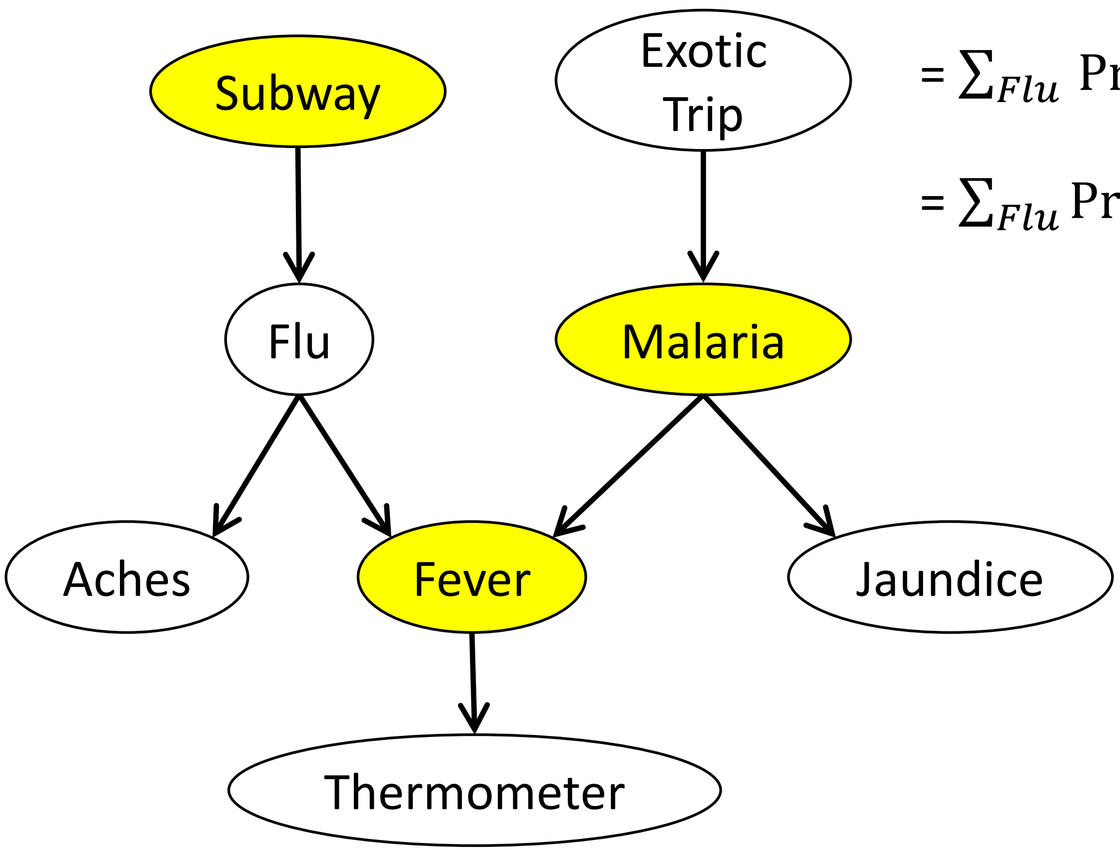
Simple Forward Inference

- Same idea applies with evidence

Pr(Fev|s,~m) = $\sum_{Flu} \Pr(Fever, Flu | s, \sim m)$

= $\sum_{Flu} \Pr(Fever | Flu, s, \sim m)$Pr(Flu|s,~m)

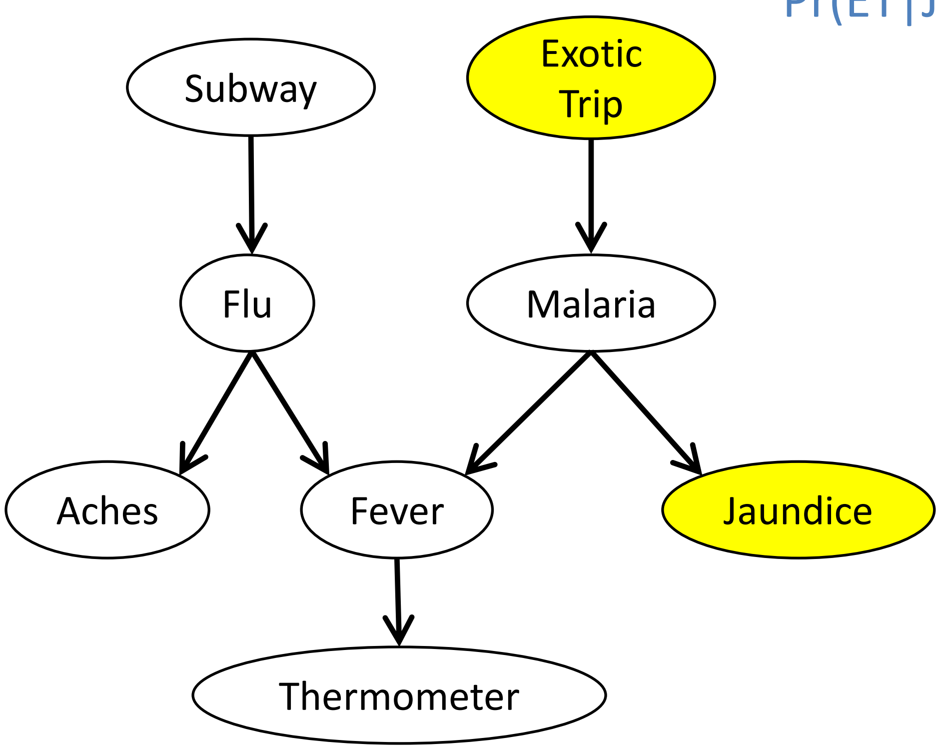
= $\sum_{Flu} \Pr(Fever | Flu, \sim m) \Pr(Flu | s)$



Simple Backward Inference

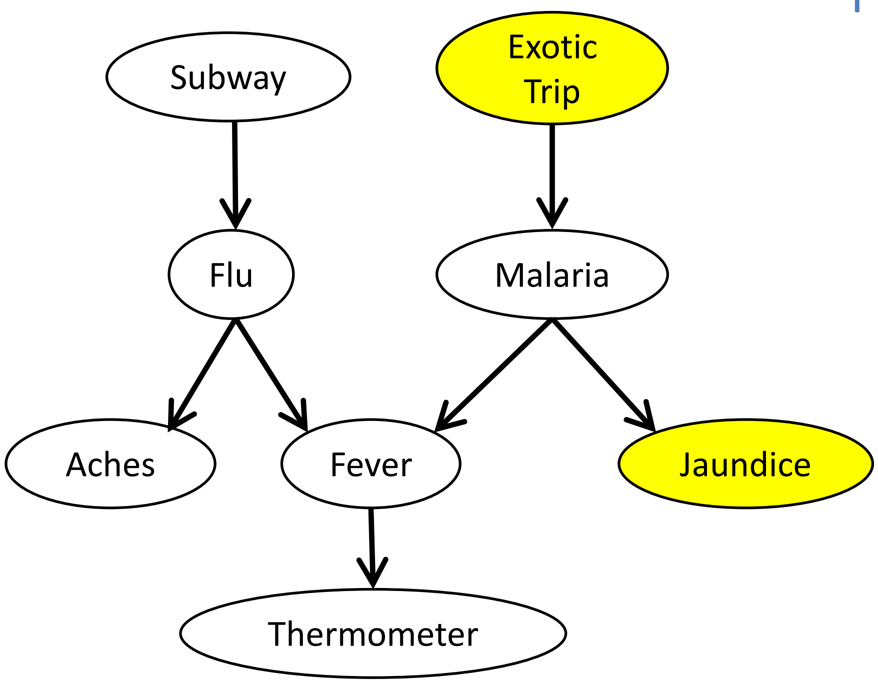
- When evidence is downstream of the query variable, we must reason "backwards"
  - Requires the use of Bayes rule

Pr(ET|J) = ?



Simple Backward Inference

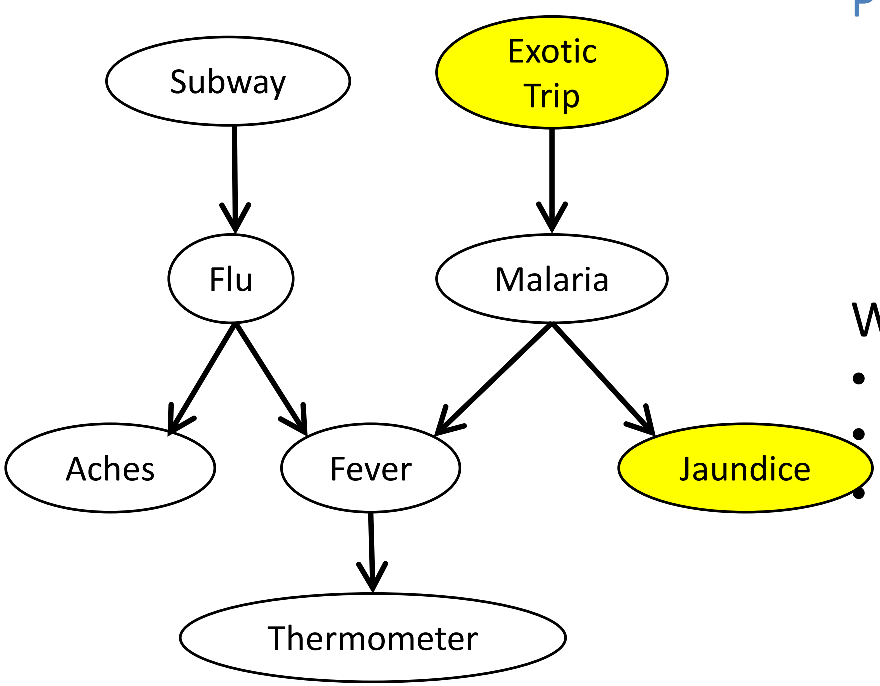
- When evidence is downstream of the query variable, we must reason "backwards"
  - Requires the use of Bayes rule



$$\Pr(ET|j) = \alpha \Pr(j|ET)\Pr(ET)$$
$$= \alpha \sum_M \Pr(j, M|ET) \Pr(ET)$$
$$= \alpha \sum_M \Pr(j|M, ET)\Pr(M|ET) \Pr(ET)$$
$$= \alpha \sum_M \Pr(j|M)\Pr(M|ET) \Pr(ET)$$

Simple Backward Inference

- When evidence is downstream of the query variable, we must reason "backwards"
  - Requires the use of Bayes rule



$$\Pr(ET|j) = \alpha \Pr(j|ET)\Pr(ET)$$
$$= \alpha \textstyle\sum_M \Pr(j, M|ET) \Pr(ET)$$
$$= \alpha \textstyle\sum_M \Pr(j|M, ET)\Pr(M|ET) \Pr(ET)$$
$$= \alpha \textstyle\sum_M \Pr(j|M)\Pr(M|ET) \Pr(ET)$$

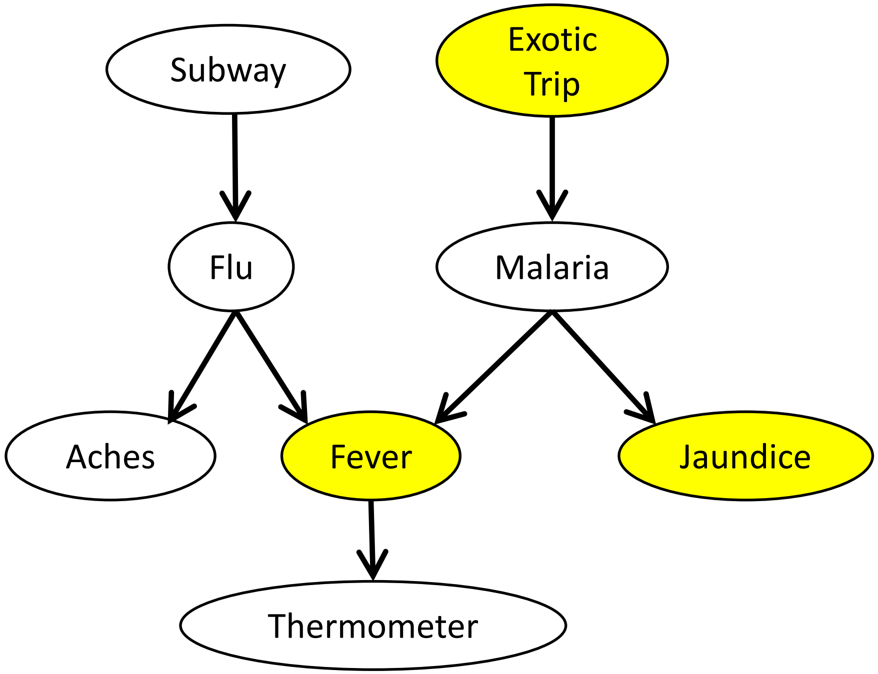
Where α = 1/Pr(j)

- Don't need to compute α
- Can compute Pr(ET|j) for each value of ET
- Add up terms Pr(j|ET)Pr(ET) for all values of ET (they sum up to Pr(j))

Simple Backward Inference

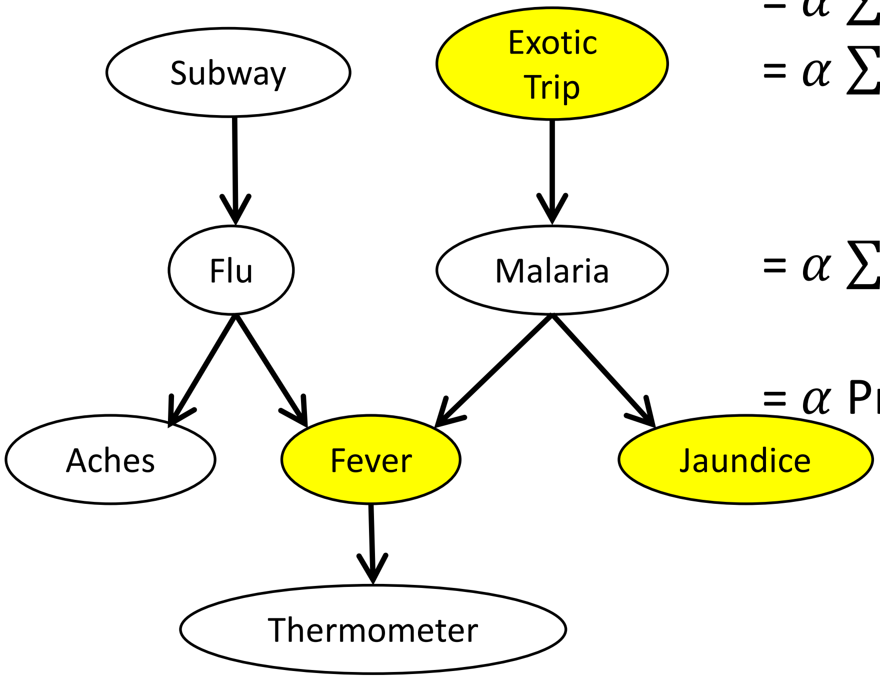
- Same idea applies when several pieces of evidence appear downstream

Pr(ET|j,fever) = ?



Simple Backward Inference

- Same idea applies when several pieces of evidence appear downstream



Pr(ET|j,fever) = ?

$= \alpha$ Pr(j,fever|ET)Pr(ET)

$= \alpha \sum_{M,Flu,S} \Pr(j, fever, M, Flu, S|ET) \Pr(ET)$

$= \alpha \sum_{M,Flu,S} \Pr(j|fever, M, Flu, S, ET)$
Pr(fever|M,Flu,S,ET)Pr(M|Flu,S,ET)
Pr(Flu|S,ET)Pr(S|ET)Pr(ET)

$= \alpha \sum_{M,Flu,S} \Pr(j|M)$Pr(fever|M,Flu)Pr(M|ET)
Pr(Flu|S)Pr(S)Pr(ET)

$= \alpha$ Pr(ET) $\sum_{M} \Pr(j|M)\Pr(M|ET)$
$\sum_{Flu} \Pr(fever|M, Flu) \sum_{S} \Pr(Flu|S) \Pr(S)$

More Efficient Algorithms

- Variable elimination
  - Push sums in as far as possible
  - Amount of work is bounded by size of largest term

More Efficient Algorithms

- Variable elimination
  - Push sums in as far as possible
  - Amount of work is bounded by size of largest term

- Dynamic programming
  - Avoid redundant computation by computing several marginals at the same time
  - If BN is a tree, can use a local message passing algorithm (called forwards-backwards algorithm in HMMs)
  - Double counts if BN has undirected cycles

More Efficient Algorithms

- Variable elimination
  - Push sums in as far as possible
  - Amount of work is bounded by size of largest term
- Dynamic programming
  - Avoid redundant computation by computing several marginals at the same time
  - If BN is a tree, can use a local message passing algorithm (called forwards-backwards algorithm in HMMs)
  - Double counts if BN has undirected cycles
- Clique/Junction tree inference
  - Converts BN into a junction tree (of clusters of nodes)
  - Then run local message passing algorithm on junction tree

More Efficient Algorithms

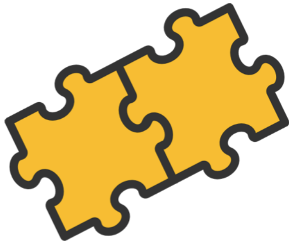
- Variable elimination
  - Push sums in as far as possible
  - Amount of work is bounded by size of largest term

- Dynamic programming
  - Avoid redundant computation by computing several marginals at the same time
  - If BN is a tree, can use a local message passing algorithm (called forwards-backwards algorithm in HMMs)
  - Double counts if BN has undirected cycles

- Clique/Junction tree inference
  - Converts BN into a junction tree (of clusters of nodes)
  - Then run local message passing algorithm on junction tree

- Approximation algorithms (sampling, loopy belief propagation, etc.)

Putting the Pieces Together



- How do we make decisions?
  - Consider each action
  - Estimate future consequences
  - Based on our beliefs of the world

- *Still need:* to model consequences to reflect how good or bad an outcome is

- Rational decision making suggests we take action with the best consequences based on our beliefs

Utility Theory

- Utilities enable us to quantify the degree of preference
  - Beyond simply ranking outcomes

- A utility function, $U: S \rightarrow \mathbb{R}$ associates a real-valued utility with each state
  - Actual values can be handcrafted or elicited/inferred
  - In economics, utilities typically correlate with monetary values

- U induces a preference ordering over S:
  s is preferred over t iff $U(s) > U(t)$
  - This preference ordering is reflexive and transitive

Expected Utility

- Under conditions of uncertainty, each decision *d* induces a distribution Pr_d over possible states
  - $Pr_d(s)$ is the probability of state *s* under decision d

- The expected utility of decision d is defined as:

$$EU(d) = \sum_{s \in S} \Pr_d(s) U(s)$$

The MEU Principle

- The principle of maximum expected utility (MEU) states that the optimal decision under uncertainty is the one with the greatest expected utility
  - Principle adopted by rational agents

- Given a set of decisions *D*, a solution to a decision problem under uncertainty is any $d^* \in D$ s.t. $EU(d^*) \geq EU(d)$, for all $d \in D$
  - Trivial to compute for single-shot problems

Tutoring Example

- Consider an intelligent tutoring system designed to assist the user in programming
- Possible actions (intended to be helpful)?

Tutoring Example

- Consider an intelligent tutoring system designed to assist the user in programming

- Possible actions (intended to be helpful):
  - A1: Auto-complete the current programming statement
  - A2: Suggest several options for completing the current statement
  - A3: Provide hints for completing the statement
  - A4: Ask if user needs help in a pop-up dialogue
  - A5: Keep observing the user (do nothing)

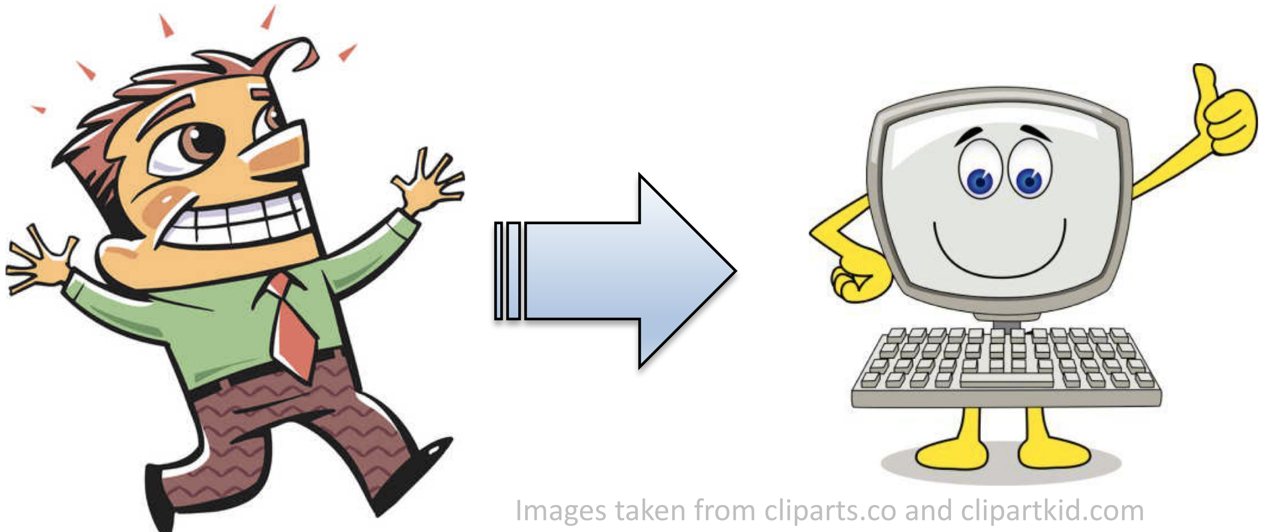
Tutoring Example

- Consider an intelligent tutoring system designed to assist the user in programming

- Possible actions (intended to be helpful):
  - A1: Auto-complete the current programming statement
  - A2: Suggest several options for completing the current statement
  - A3: Provide hints for completing the statement
  - A4: Ask if user needs help in a pop-up dialogue
  - A5: Keep observing the user (do nothing)

- Every action has consequences (expressed as utilities)
  - their impact depend on how much help the user needs right now

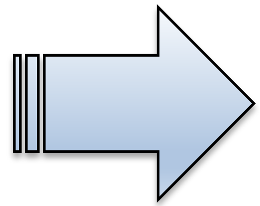
“Your Happiness is My Happiness”



Images taken from cliparts.co and clipartkid.com

- System acts to help user
- If user is happy, system is doing the right thing

“Your Happiness is My Happiness”



Images taken from cliparts.co and clipartkid.com

- System acts to help user
- If user is happy, system is doing the right thing

- In this case:
  - System makes actions to keep user happy
  - Utility function should reflect user’s preferences

Utility Function with User Variables

- System's decision problem:
  - Which is the best action to take to make user most happy?
  - With consideration to how much help the user needs right now

Utility Function with User Variables

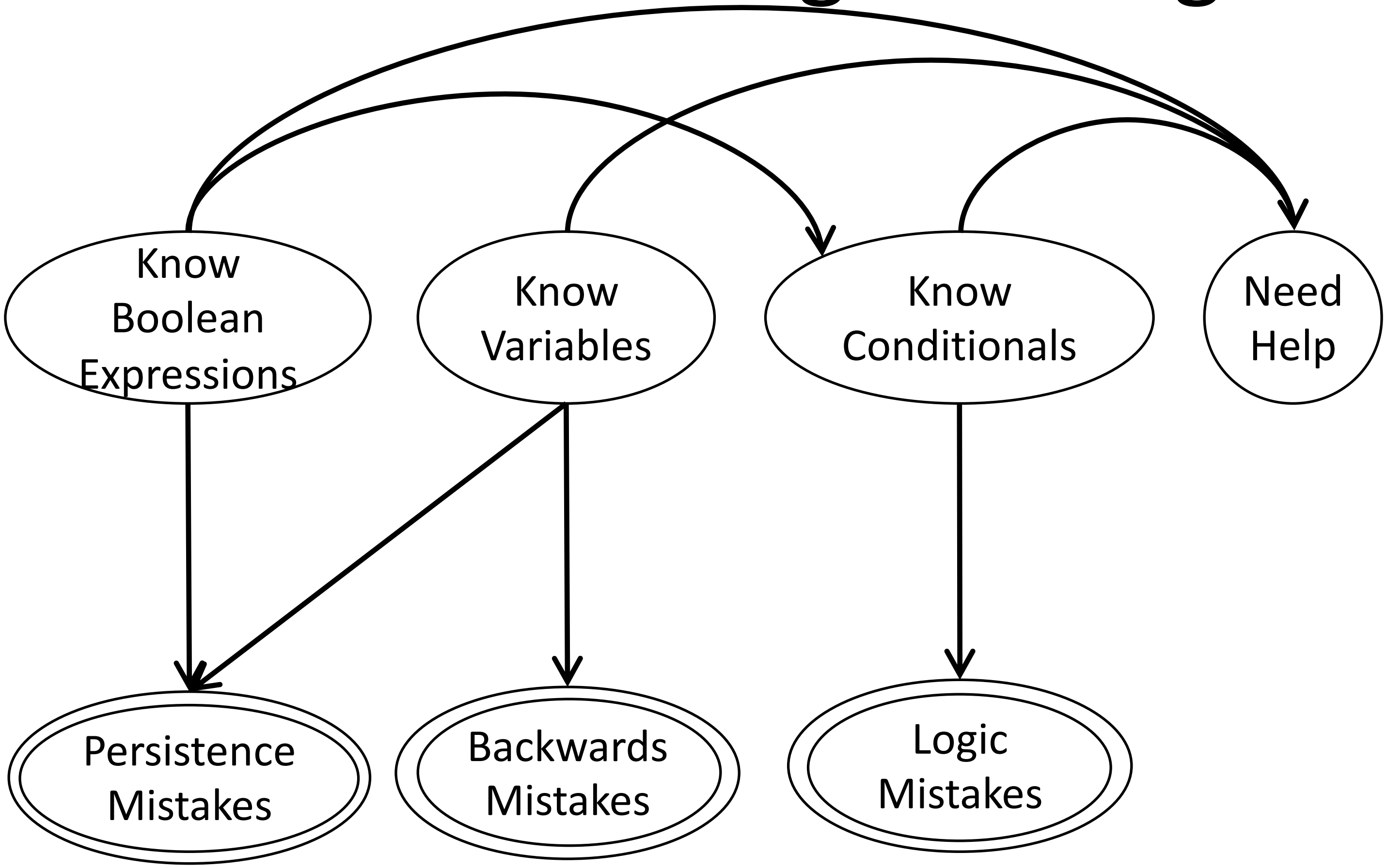
- System's decision problem:
  - Which is the best action to take to make user most happy?
  - With consideration to how much help the user needs right now

- EU(Action) = Pr(NeedHelp) × U(NeedHelp,Action)

Comes from marginal distribution computed from Bayes net

Comes from utility defined/elicited and stored separately

Recall Model for Programming Help



Compute Pr(Help) using model

Defining U(NeedHelp,Action)

- Define U in a fixed range e.g. [0,100] or [-5, +5]

- For each scenario, assign real number to each action: auto-complete, suggest options, hint, ask, do nothing

- If NeedHelp is high,
  - Auto-complete, Suggest options, Hint are more appropriate
  - Note: Should also model quality of suggestion
- If NeedHelp is medium,
  - Ask and Hint is more appropriate
- If NeedHelp is low,
  - Do nothing is more appropriate

- Compute EU of each action and take the action with highest EU

Key Ideas

- Main concepts
  - Simple forward and backward inference involves basic probability manipulations and calculations
  - Rational decision making adopts the maximum expected utility principle
- Representation:
  - Prediction – given evidence upstream, predict likely outcome
  - Explanation – given evidence downstream, find plausible cause
  - Multiple causes compete to explain evidence away
  - Utility function $U: S \rightarrow \mathbb{R}$ is a mapping of states to reals
- Main tasks of interest:
  - Updating belief of **X**: $\Pr(\mathbf{X}|\mathbf{e})$
  - Most probable explanation: $\mathbf{x}^* = \text{argmax}_{\mathbf{x}} \Pr(\mathbf{x}|\mathbf{e})$
  - Expected utility of a decision: $EU(d) = \Sigma_s \Pr_d(s) U(s)$
- Algorithm:
  - Most general exact inference algorithm is called clique (junction) inference

Further Readings

- D. Warner North. A Tutorial Introduction to Decision Theory. IEEE Transactions on Systems, Science, and Cybernetics, 4(3), 1968.
- S. French. Decision Theory. Halsted, 1986.
- S. Russell and P. Norvig. AIMA: Chapter 16.